Hochschule Konstanz
Fakultät Informatik

Machine Learning:: Model Evaluation

Dozenten: Prof. Dr. M. O. Franz, Prof. Dr. O. Dürr

Remark: Much of the material have been developed together with Beate Sick (ZHAW/Uni Zurich).

Bayes-Klassifikator

Für einen gegebenen Meßwert x schätzt der Bayes-Klassifikator die Klassenzugehörigkeit nach der

Bayessche Entscheidungsregel

Entscheide Dich immer für die Klasse ω_i, deren A-posteriori-Wahrscheinlichkeit $p(\omega_i|x)$ am höchsten ist.

Der Bayes-Klassifikator ist (theoretisch) der bestmögliche Klassifikator, da er die Wahrscheinlichkeit einer Fehlklassifikation minimiert. Jeder andere Klassifikator macht mehr oder mindestens gleich viele Fehler.

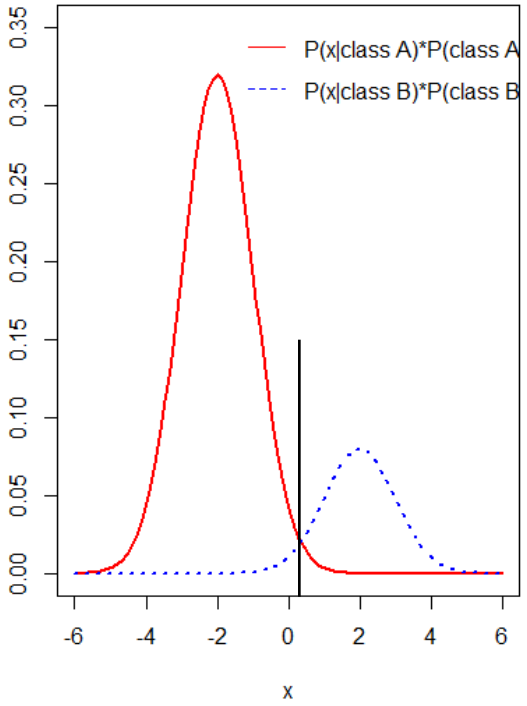
Je nachdem, welche A-posteriori-Wahrscheinlichkeit am größten ist, wird der Inputraum in Entscheidungsregionen aufgeteilt. Die Entscheidungsregionen sind durch Entscheidungsgrenzen voneinander getrennt.

Summary Bayes Classifier

Each class has a prior probability P(A) and P(B)
⟹before seeing data we classify to class with higher prior.

Then we observe feature value x and determine the posterior probabilities P(A|x) and P(B|x)
⟹ we then classify to class with higher posterior probability

We use the Bayes theorem to determine the posterior:



$$P(\text{class} = C \mid \text{X} = \text{x}) = \frac{P(X = x \mid \text{class C}) \cdot P(\text{class C})}{P(X = x)} \quad , \quad \text{C} \in \{\text{A,B}\}$$

With $P(X = x) = P(X = x \mid \text{class A}) \cdot P(\text{class } A) + P(X = x \mid \text{class B}) \cdot P(\text{classB})$

Decide for class A if: $P(\text{class} = A \mid \text{X} = \text{x}) > P(\text{class} = B \mid \text{X} = \text{x})$

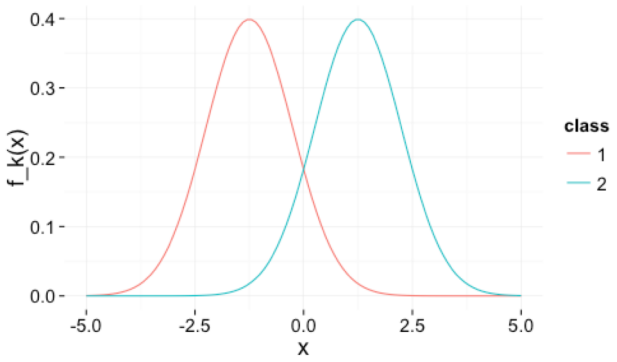
$$\Leftrightarrow \frac{P(X = x \mid \text{class A}) \cdot P(\text{class A})}{P(X = x)} > \frac{P(X = x \mid \text{class B}) \cdot P(\text{class B})}{P(X = x)}$$

$$\Leftrightarrow P(X = x \mid \text{class A}) \cdot P(\text{class A}) > P(X = x \mid \text{class B}) \cdot P(\text{class B})$$

Signal Detection Theory

Accuracy of the Bayes Classifier

Aufgabe



$$X_{class\_1} \sim N(\mu_1 = -1.25, \sigma^2 = 1)$$

$$X_{class\_2} \sim N(\mu_2 = +1.25, \sigma^2 = 1)$$

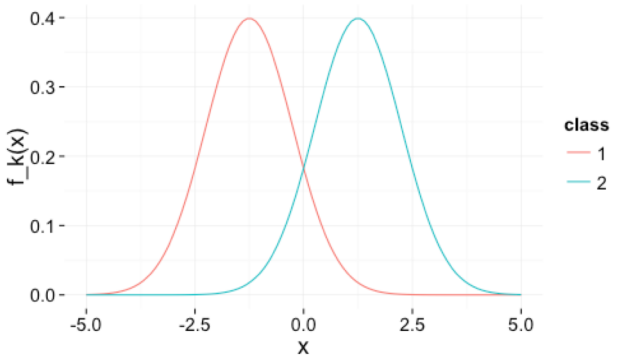
$$\int_{-\infty}^{0} Norm(x, \mu = -1.25, sd = 1)dx = 0.89$$

from scipy.stats import norm
norm.cdf(0, -1.25)
0.89

		PREDICTED CLASS	
		Class=1	Class=2
ACTUAL CLASS	Class=1		
	Class=2		

Accuracy of the Bayes Classifier

Lösung



$$X_{class\_1} \sim N(\mu_1 = -1.25, \sigma^2 = 1)$$

$$X_{class\_2} \sim N(\mu_2 = +1.25, \sigma^2 = 1)$$

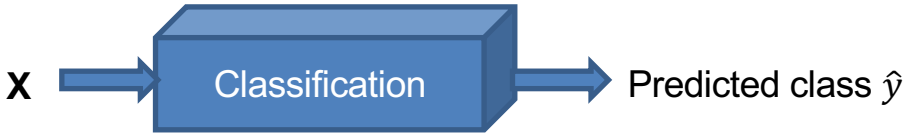
	PREDICTED CLASS		
ACTUAL CLASS		Class=1	Class=2
	Class=1	89%	11%
	Class=2	11%	89%

Without Proof: Bayes decision rule yields optimal accuracy.

Signaldetektion
ROC Analysis for Score Based Classifiers

Confusion matrix: Evaluate classification performance

Evaluation is done on a test set with known true class y and the predicted class $\hat{y}$.



id	true_class	pred_class
1	P	P
2	N	P
3	N	N
4	P	P
5	N	N
6	N	N

		True class: Positive	True class: Negative
Predicted class	Positive	TP=2	FP=1
	Negative	FN=0	TN=3

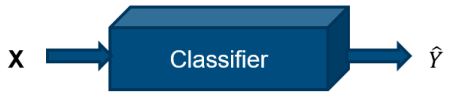
Operating on different levels of trust (motivation)

- Example (from ILSR, chapter 4.4.3), credit score prediction.
  - Goal is predict if a credit will be defaulted.
  - Classifier X makes 252+ 23 mistakes on 10000 predictions (2.75% misclassification error rate).

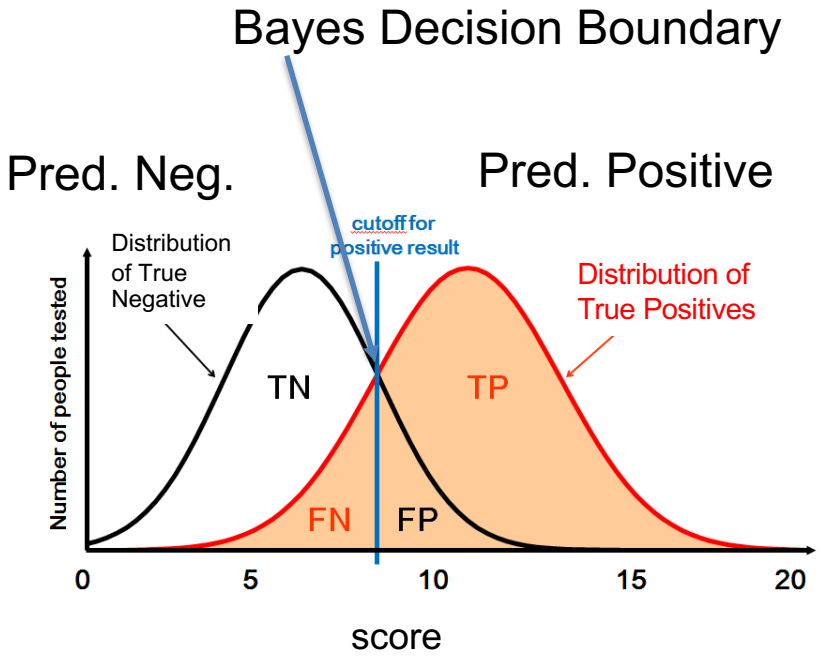
	True Default: yes	True Default: non	Total
Pred Default: yes	81 (TP)	23(FP)	104
Pred Default: no	252 (FN)	9644 (TN)	9896
Total	333	9667	10000

- **Is this a good classifier?**
- **Do you have an idea of a simple classifier with misclassification rate 3.33%?**
- **Change the decision boundary** (e.g. away from bayes)

Score based classifier



continous score*



- Output: continuous score $\hat{Y}(x)$

  (instead of class prediction)
- Discretized by choosing a cut-off
  - score ≥ c ➔ class «positive» or 1
  - score < c ➔ class «negative» or 0

		True class: Positive	True class: Negative
Predicted class	Positive	TP	FP
	Negative	FN	TN

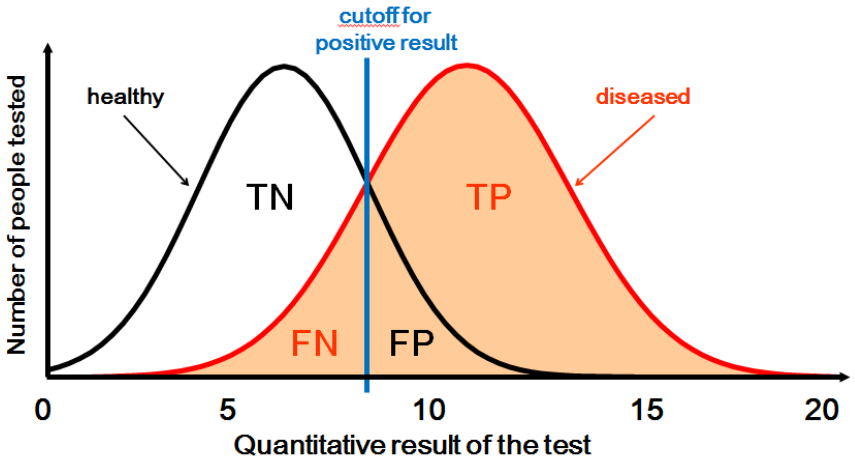
*Examples: all probabilistic classifiers (e.g. log regression), Bayes. Also others like SVM (distance to hyperplane)

Operating on different levels of trust (motivation)

- Classify more as defaulters
- Now the total number of mistakes is 235+138 = 373 (3.73% misclassification error rate)
- But we only miss-predicted 138/333 = 41.4% of defaulters
- We can examine the error rate with other thresholds

	True Default: yes	True Default: non	Total
Pred Default: yes	195(TP)	235(FP)	430
Pred Default: no	138 (FN)	9432 (TN)	9570
Total	333	9667	10000

Decision errors in a 2 class problem



Confusion matrix:

predicted class \ true class	1	0
1	True Positive	False Positive
0	False Negative	True Negative

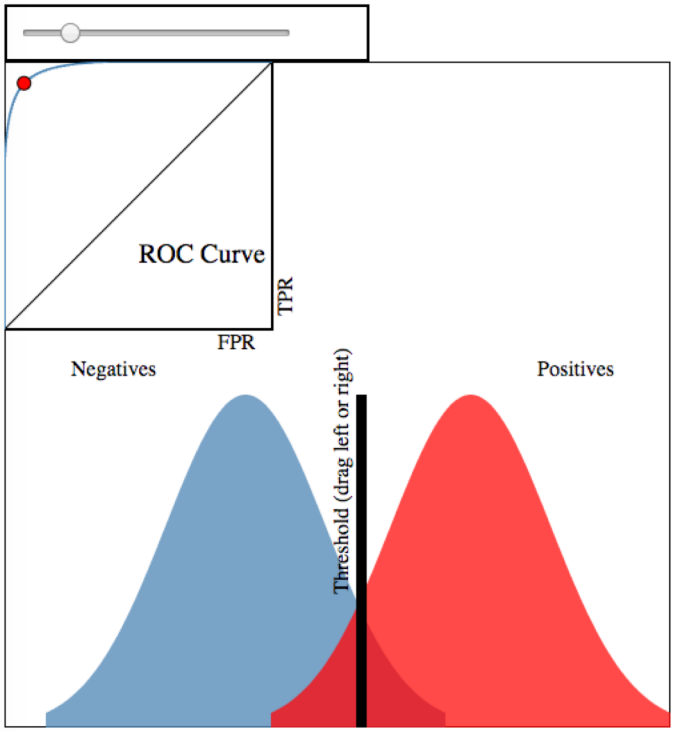
The true positive rate (sensitivity) of classifier is the ability to identify correctly the class 1 defaulters

P(TP)=TP / (TP + FN)

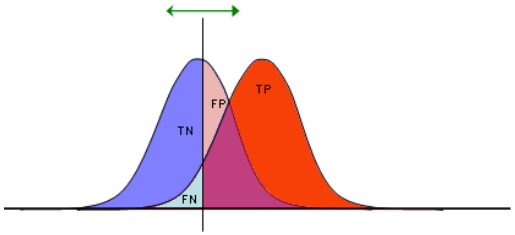
The false positive rate are the ones acussed wrongly

P(FP)=FP / (FP + TN)

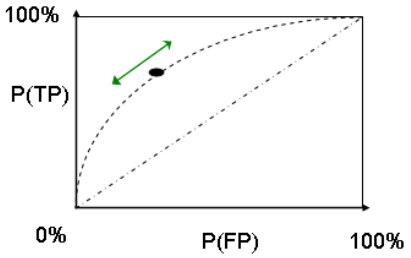
Understanding ROC-Curves



http://www.navan.name/roc/



TP	FP
FN	TN
1	1



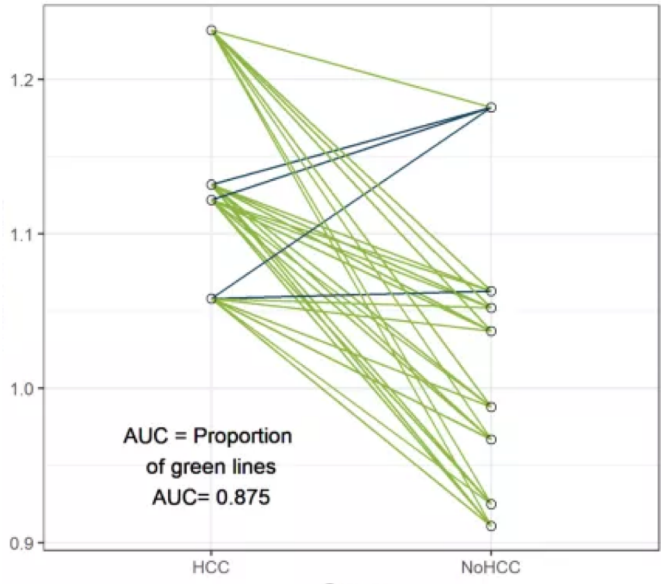
taken from Wikipedia

See also http://www.dataschool.io/roc-curves-and-auc-explained/ for an introduction to ROC

Notes on AUC (alternative interpretation)

- There is an alternative interpretation (not to well kown):
  - The AUC of a classifier is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance
  - To estimate it:

Score e.g. prob
For class 1



Class 1 Class 2

4*8 = 32 Pairs

Correct/all = 28/32=0.875

https://www.r-bloggers.com/interpretation-of-the-auc/

Notes on AUC

- Nice Properties of the AUC
  - Independence of the decision threshold.
  - Invariance to prior class probabilities or class prevalence in the data.
    - Especially usefull for rare events

- Further reading (How to get confidence intevalls)
  - https://blog.revolutionanalytics.com/2017/03/auc-meets-u-stat.html

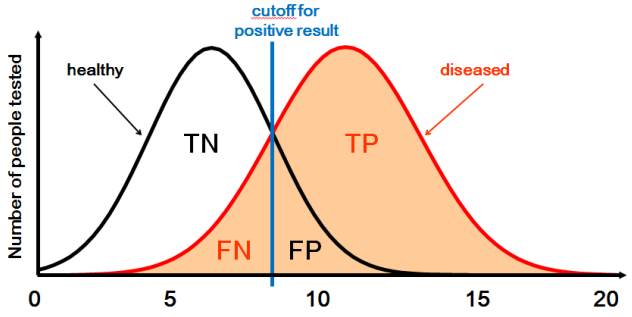
Summary: Types of Classifiers

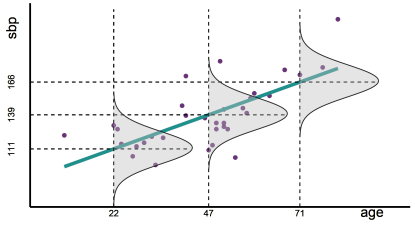
- Deterministic: Assigns X to a class
  - Examples: Perceptron
- Score based: Assigns X to a number
  - The higher that number the more “sure” is the predictions
  - Examples: SVM (to come)
- Probabilistic Classifiers: Assigns X to a number and the number is a probability
  - Examples: Logistic Regression, Bayes (theoretical), Naïve Bayes

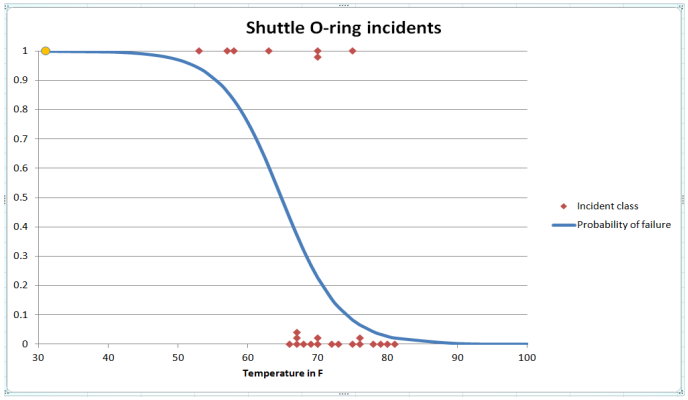
Summary: Performance Measures for Classifiers

- Deterministic Classifier
  - Confusion Matrix Based
    - Accuracy
    - Precision, Recall, F1 Score, …
- Score based classifiers
  - ROC
    - AUC and various others
- Probabilistic classifiers
  - Likelihood
    - Negative Log Likelihood (per example)
  - Likelihood is not limited to classification

		True class: Positive	True class: Negative
Predicted class	Positive	TP	FP
	Negative	FN	TN







Crossvalidation

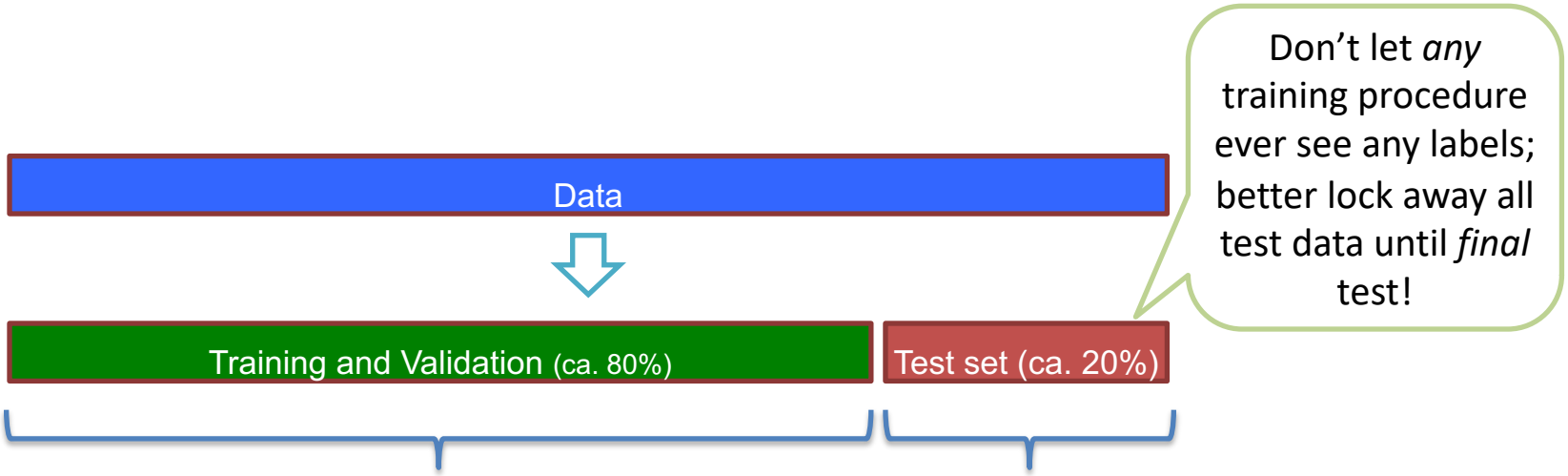
Questions for X-Validation

- Model Selection (which model to take)
  - Often there is a knob controlling the complexity of a classifier
    - Number of NN in KNN?
  - Number of features to use?
  - Square / log the features and add them?
  - Shall I use Logistic Regression or Perceptron?

- Model Evaluation
  - How good is the performance on new unseen data.
    - Note that the performance can also be
      - AUC instead of Accuracy
      - Likelihood of observations instead of AUC or accuracy

How to be on the safe side

Typical strategy, spare some data for the final testing.

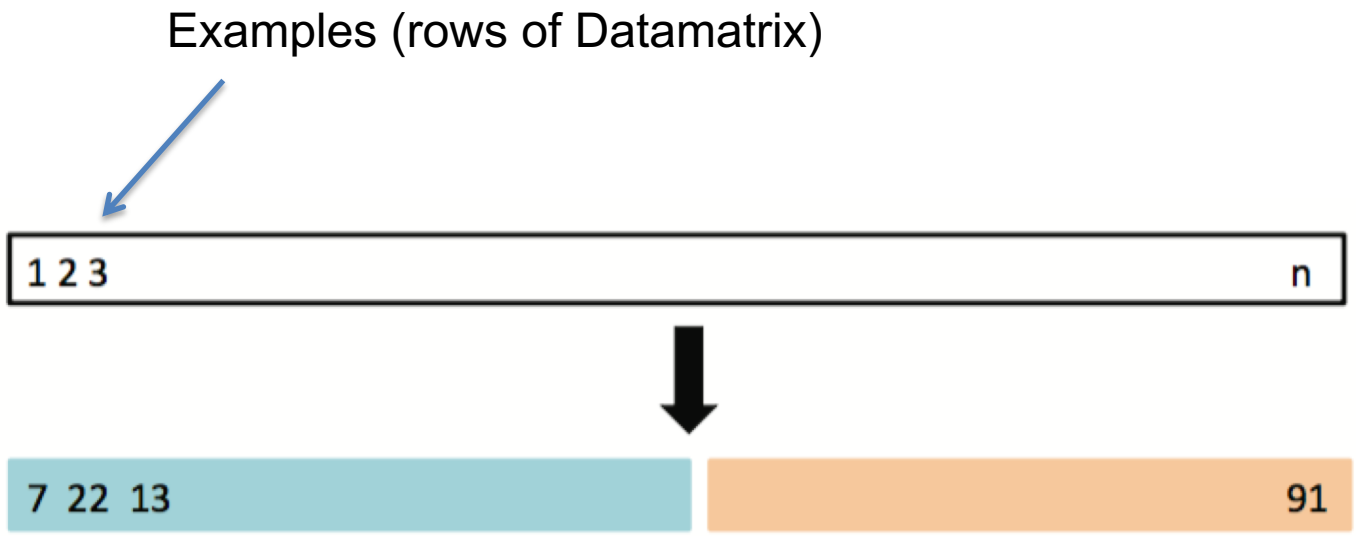


Use this to train model, select models and hyperparameters

Use this to estimate the performance.

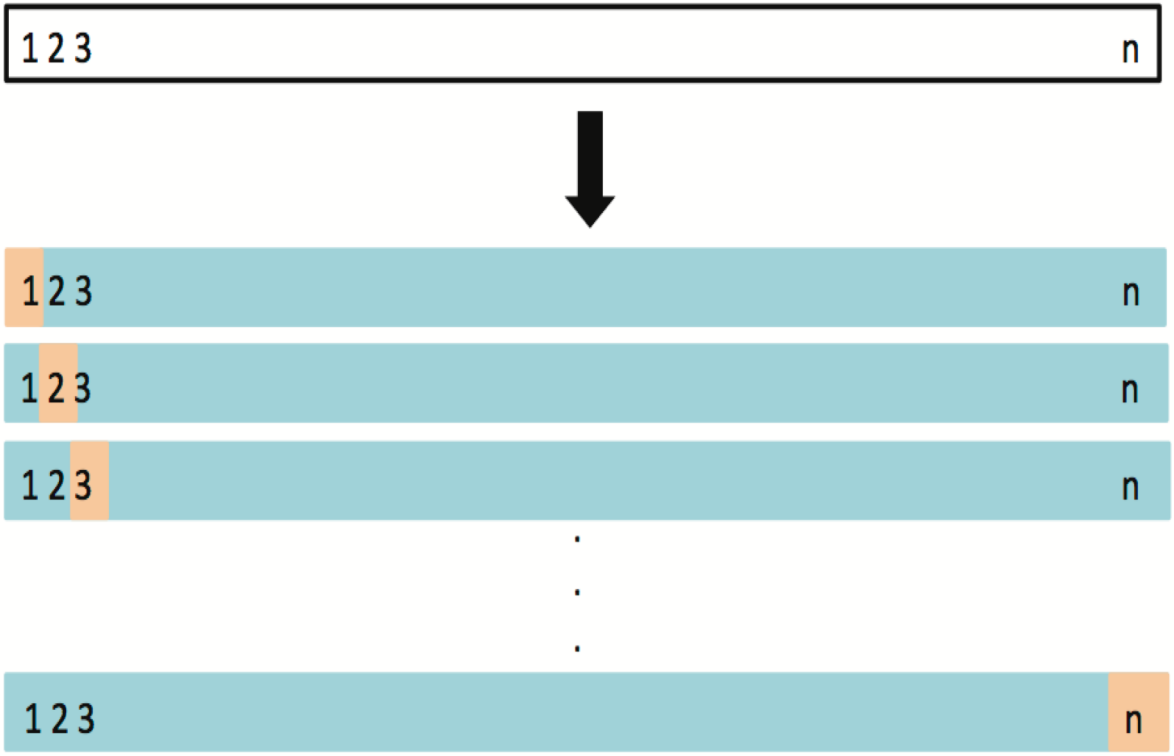
In the following, we are only talking about Training and Validation set.

The Validation Set Approach



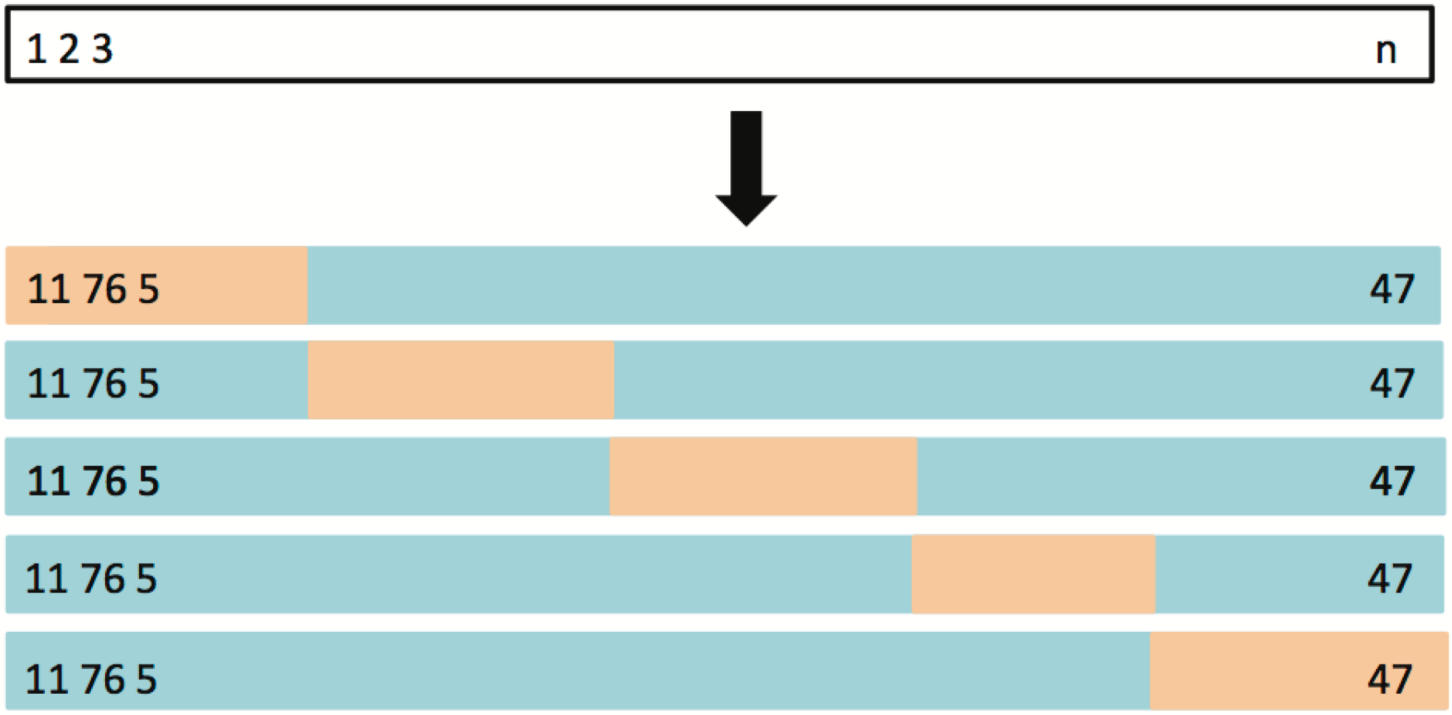
A random splitting into two halves: left part is training set, right part is validation set

Leave-One-Out Cross Validation (LOOCV)



Fit w/o red sample and predict the red sample. Average over all n repeats

K-fold Cross Validation



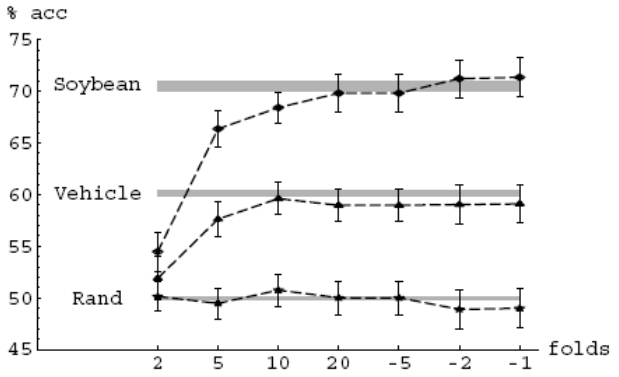
Fit w/o red samples and predict the red samples. Average over all k repeats. Do a weighted average if folds do not have the same size.

Question: What happens if k=n, what if k=2?

What is the best cross-validation scheme?

- LOOCV
  - Not random
  - Sometimes slow (there are build in procedures for some classifiers)
  - Higher Variance (between completely new data sets)
  - Lowest possible bias (nearly the whole data seen)

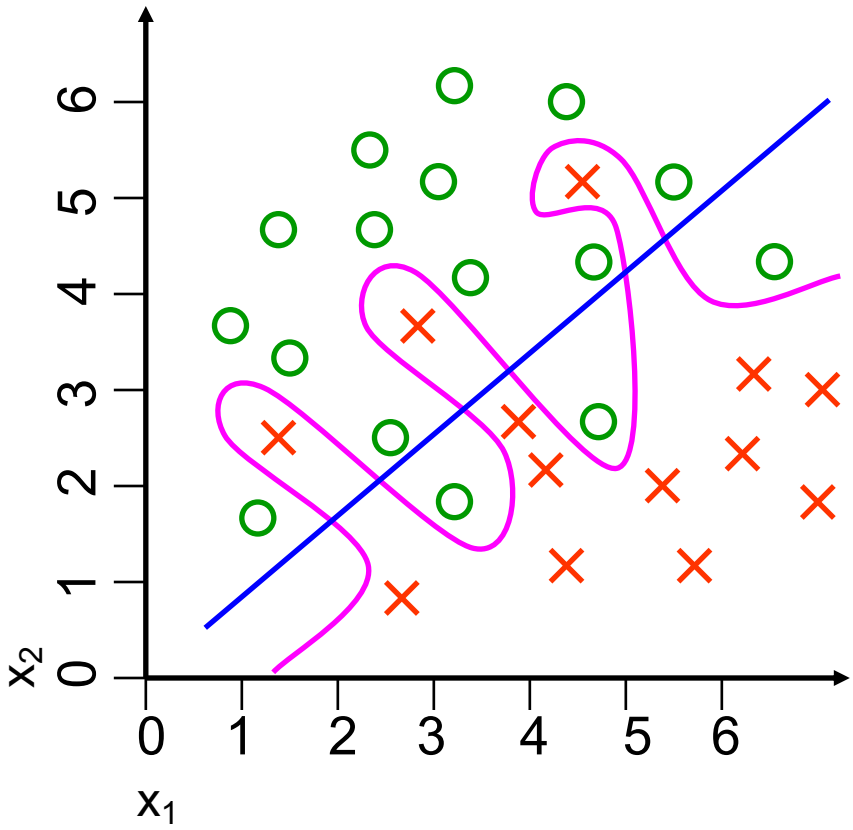
- K-Fold Xvalidation
  - Random result
  - Usually faster
  - Lower Variance
  - Higher Bias (usually not problematic)



Standard today: k=10 Fold cross-validation (sometimes repeated after permutation)

Bias Variance Tradeoff

"Perfect" Vs. "Simple" classifier



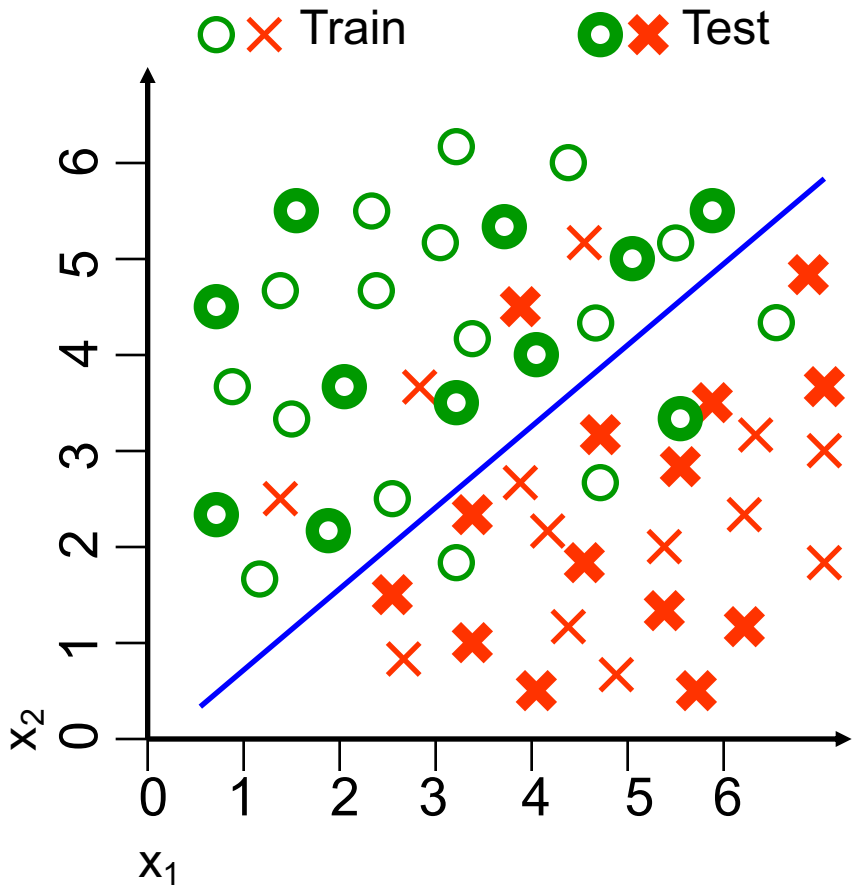
Which is better?

Check on a test-set (don't use all you labeled data to train)

Cross validation of the "simple" classifier

Training set:
6/29=20%
misclassification

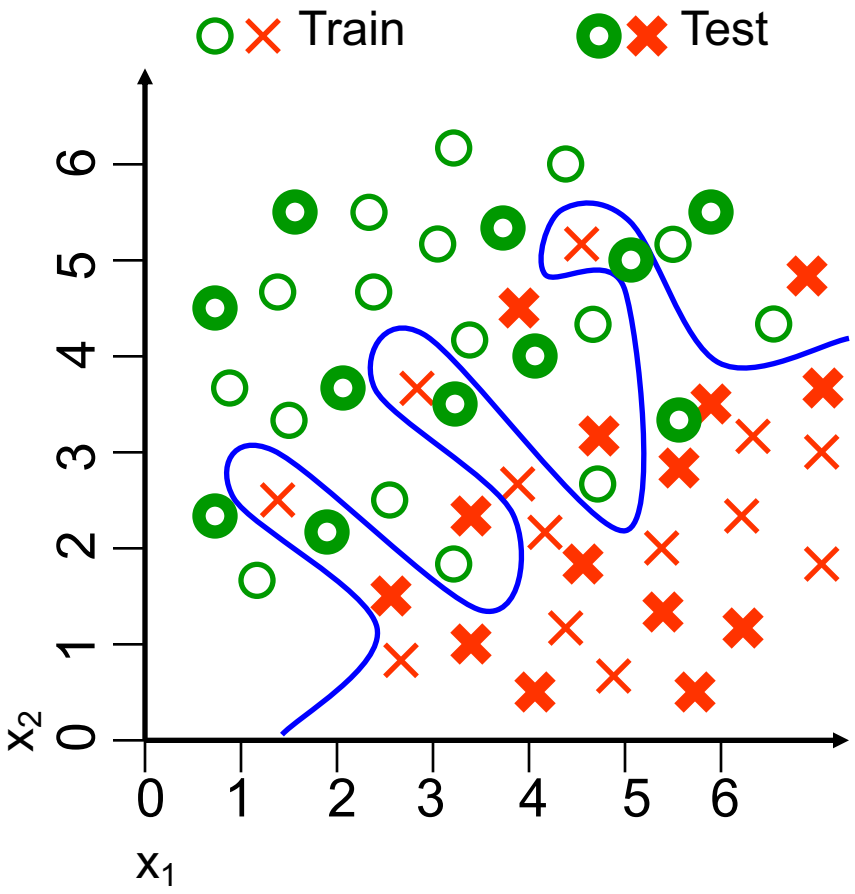
Test set:
2/25=8%
misclassification



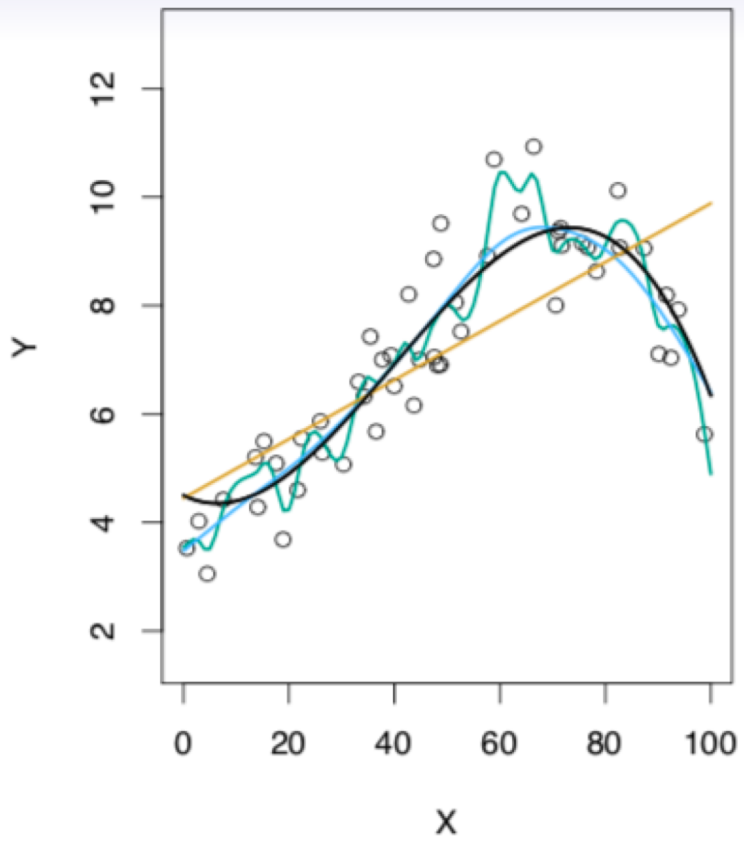
Cross validation of the "Perfect" classifier

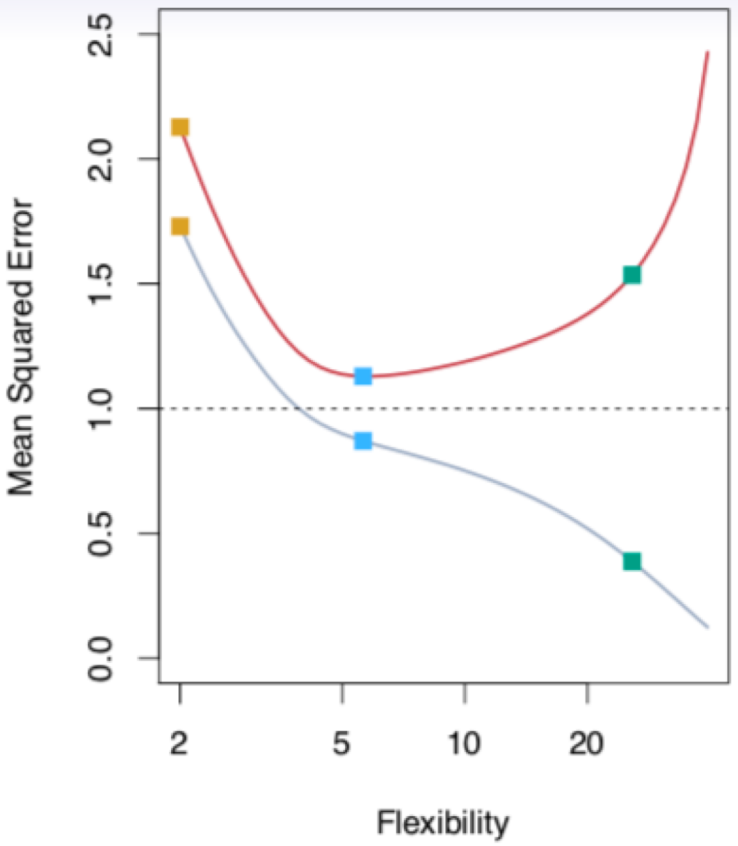
Training set:
0%misclassification

Test set:
8/25=24%
misclassification



Some examples from regression

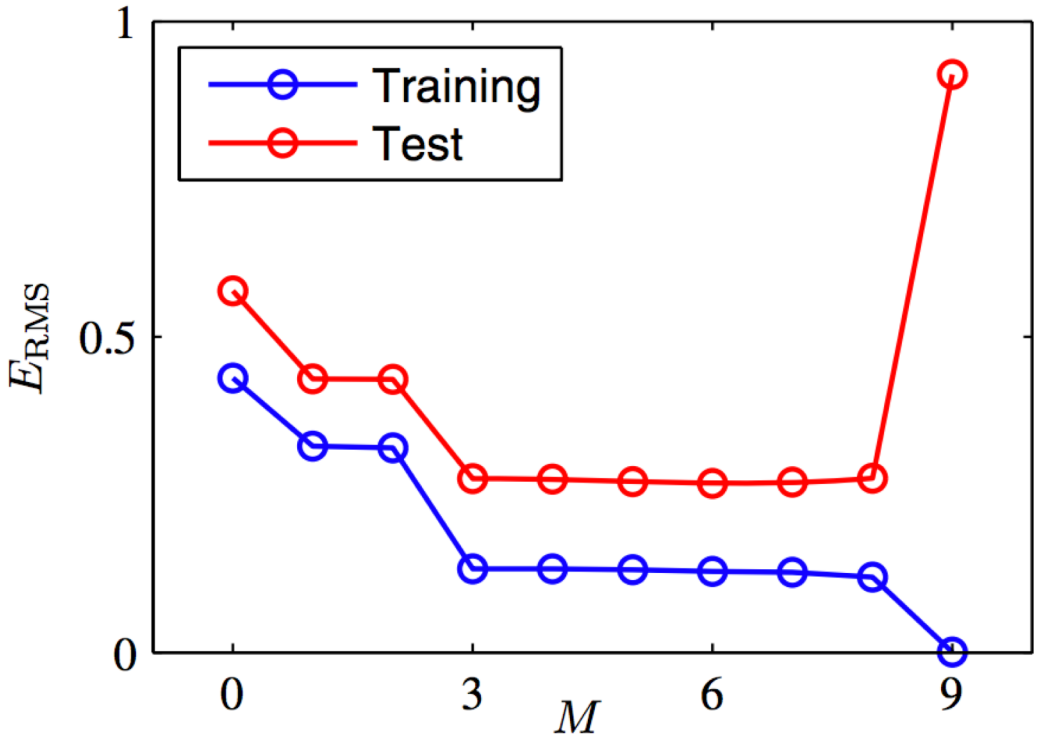




Black curve is truth. Red curve on right is MSE_{Te}, grey curve is MSE_{Tr}. Orange, blue and green curves/squares correspond to fits of different flexibility.

Taken from ILSR

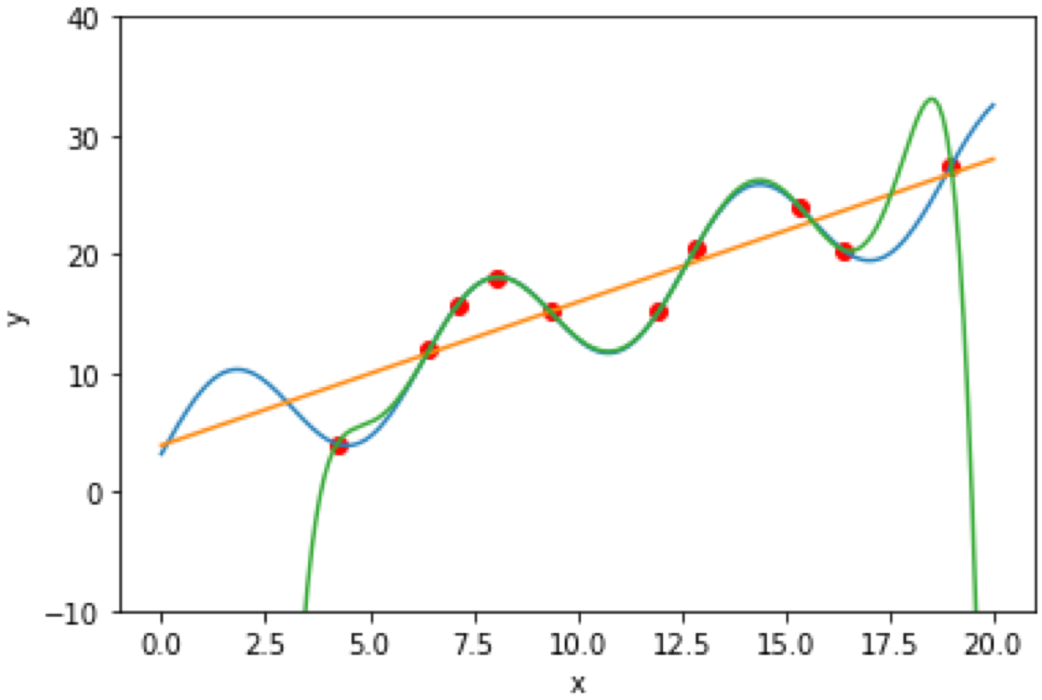
Over- and Underfitting: Intuition



Too simple model
Underfitting

Too complex model
Overfitting

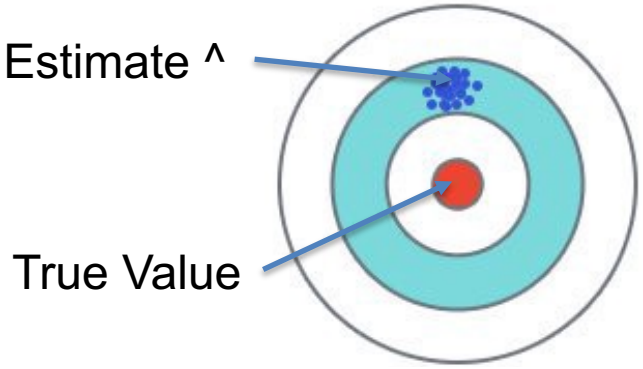
Which model will yield the better predictions?



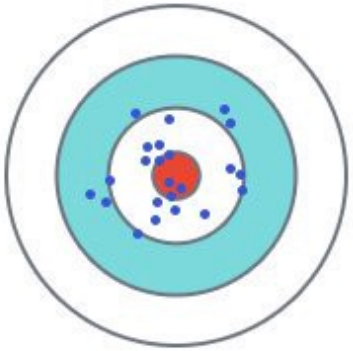
Demo Time

Simulation study: https://github.com/ioskn/mldl_htwg/blob/master/bias_variance/Bias_Variance_Sim.ipynb

Two possible ways to be wrong (bias vs. variance)



Systematically off
(Bias)



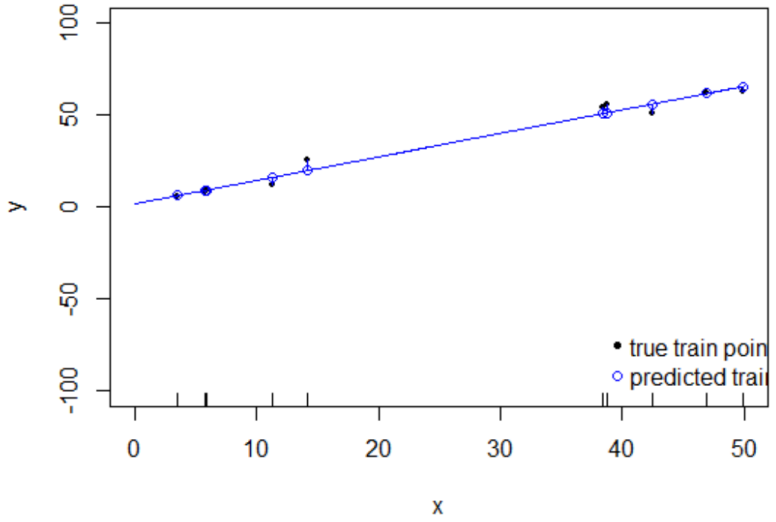
Large fluctuations
(Variance)

Bias and variance of estimates (stuff with hat) for any estimator for example:

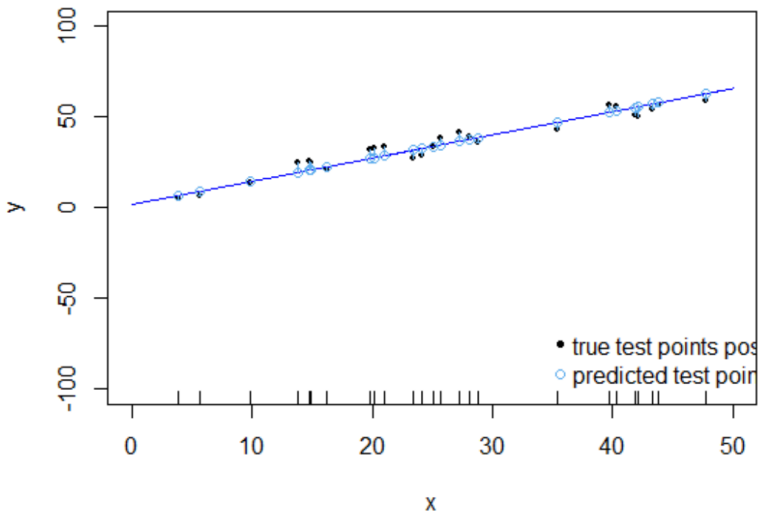
- The prediction $\hat{f}(X)$ compared to $f(X)$
- The parameter $\hat{\beta}$ compared to β

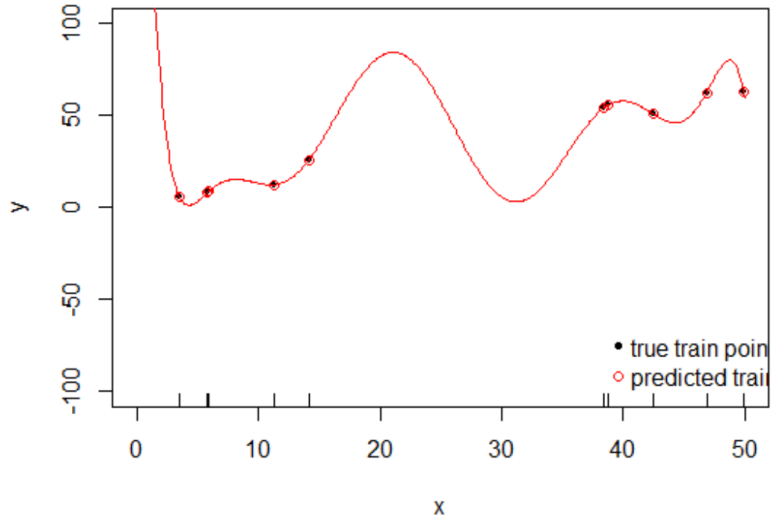
Compare flexible with ridge prediction model

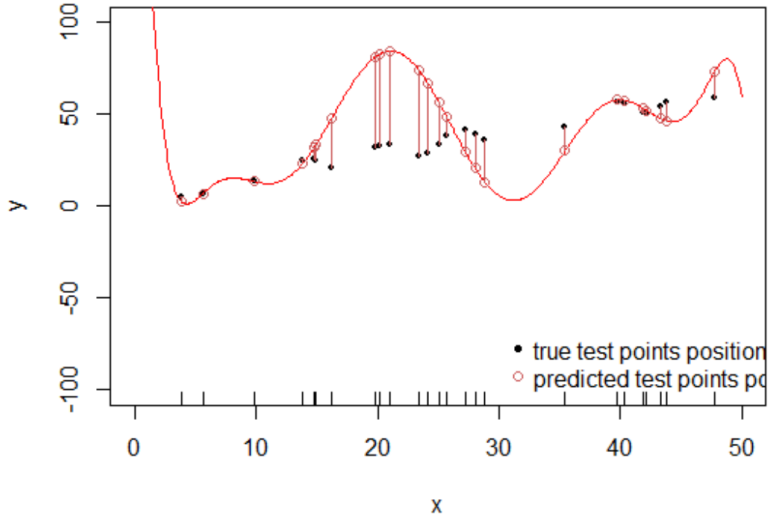
Check performance on **train** data



Check performance on **test** data





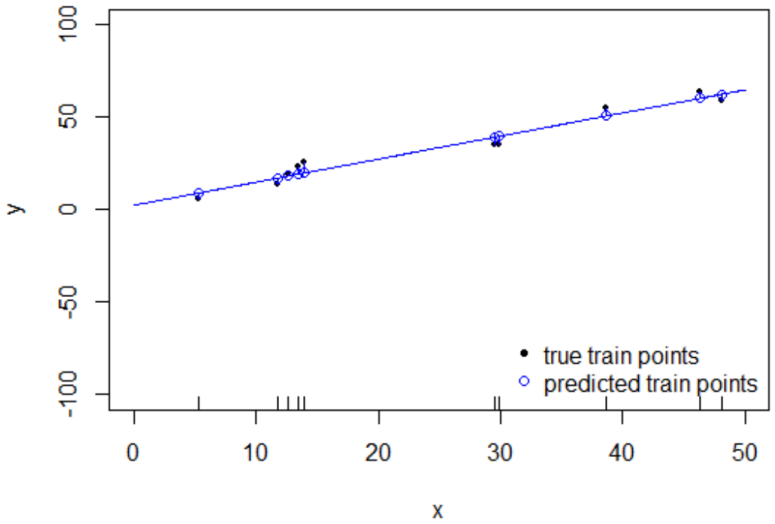


Note: The flexible “overfits” the train data: its performance goes down on test data

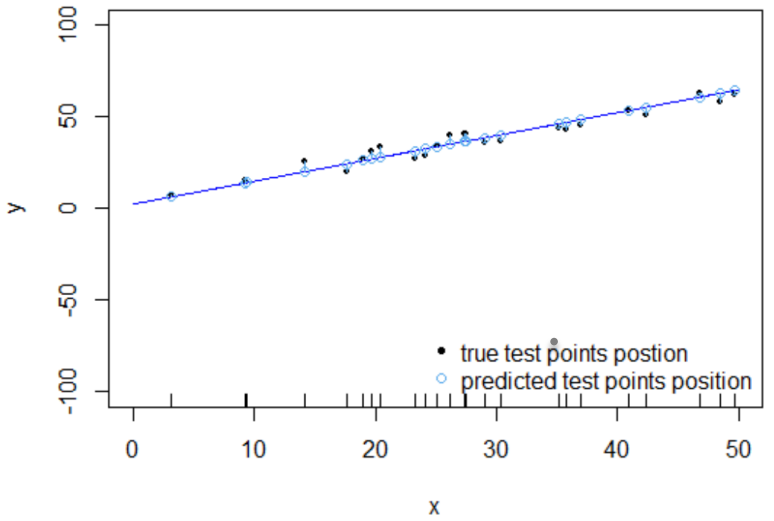
The rigid model often “underfits” the data: the true underlying relationship is more complex.

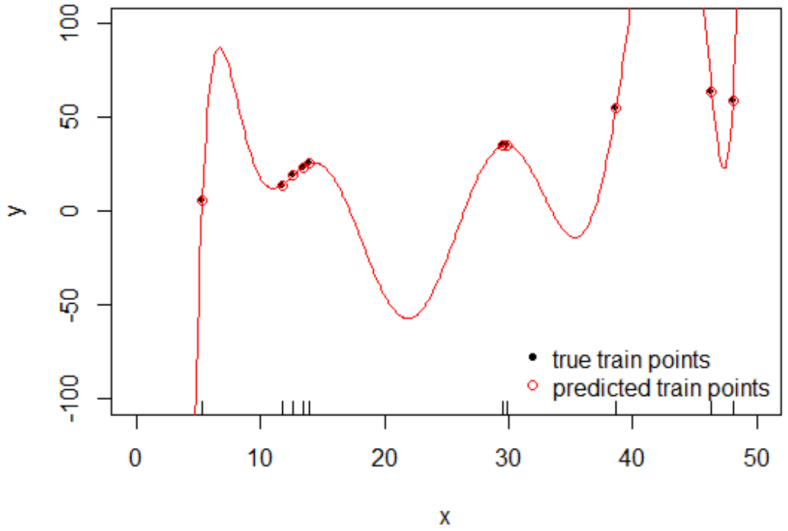
Get new train and test data and do all again

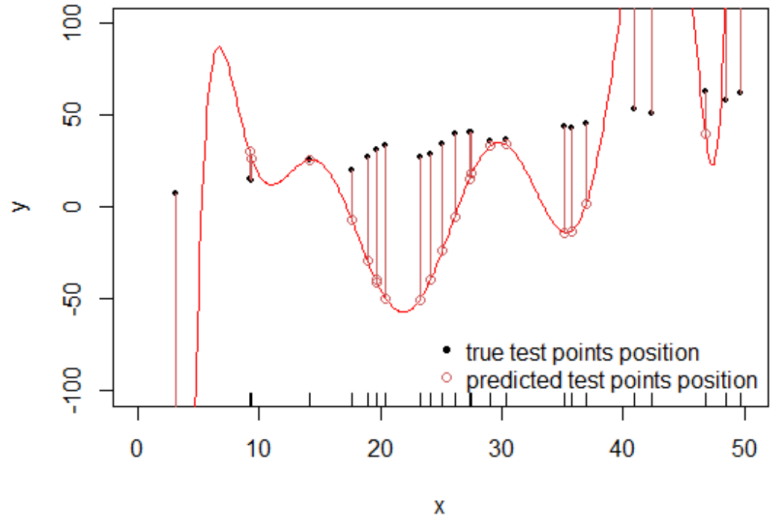
Check performance on **train** data



Check performance on **test** data



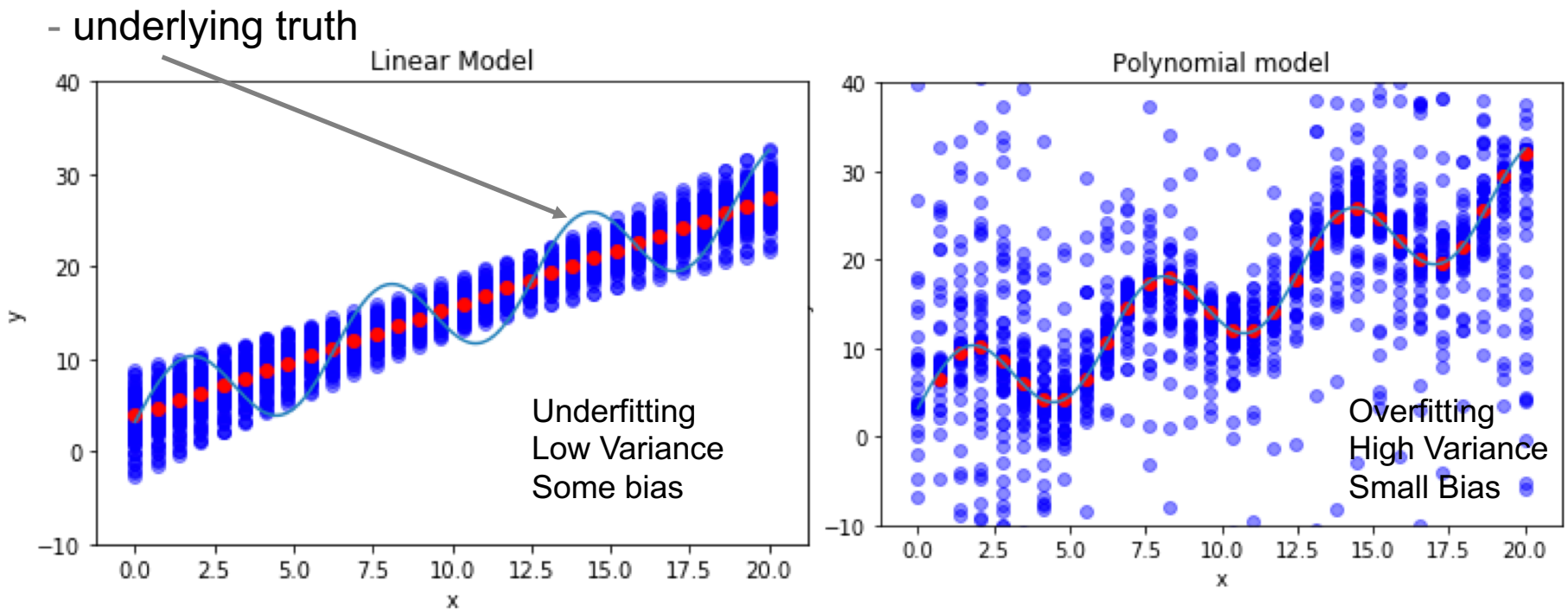




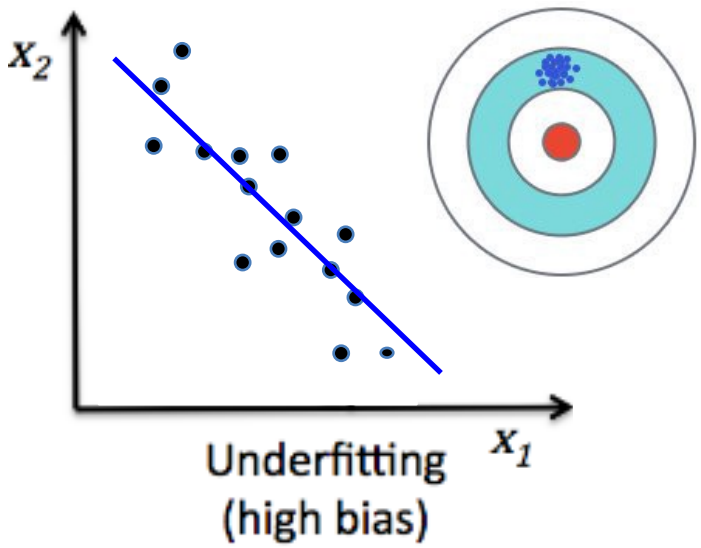
Note: The flexible model is on new train data very different to the run we saw before, the rigid model is very simil

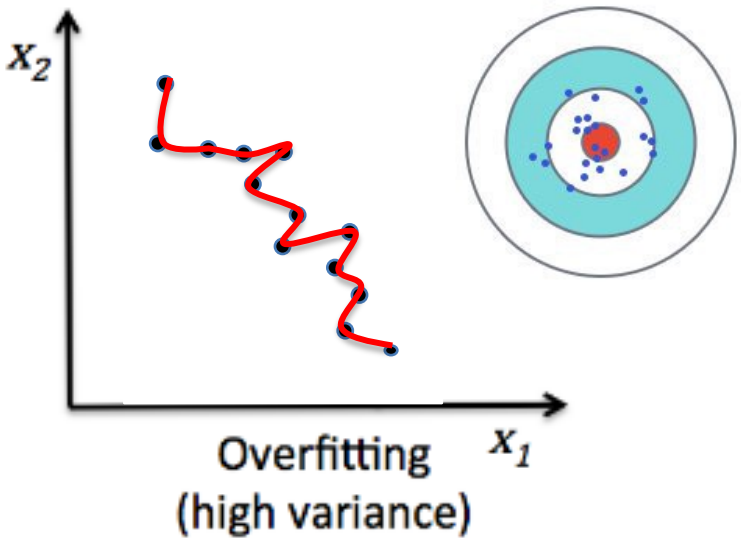
Variance/Bias trade-off of flexible/rigid models

- Simulated data came from an ascending oscillating curve – see grey curve.
- 200 Runs
  - Sample Train set of 10 points and fit both models
  - Evaluate test set at 30 pre-selected x-positions (see tiny points) mean fat points.



The concept of Bias and Variance





A under fitting model

- Is **not flexible enough** for true data structure
- Some systematic error since the model assumes a too simple relationship (high bias)
- Will **not vary a lot if fitted to new train data** (low variance)

A overfitting model

- Is **too flexible for data structure**
- Few errors on train set
- Non-systematic test errors (low bias)
- Will **vary a lot if fitted to new train** data (high variance)

Later Ensemble Methods will use this inside